Enhanced On-line authentication using Zero-knowledge protocol based on Virtual password

Major Project Report Submitted In Partial Fulfillment Of The Requirements
For The Award Of The Degree Of

MASTER OF TECHONOLOGY IN COMPUTER SCIENCE

By
VENU MANGA (08MCMT22)

DEPARTMENT OF COMPUTER AND INFORMATION SCIENCES
UNIVERSITY OF HYDERABAD
HYDERABAD
APRIL-2010

Department of Computer &Information Sciences
School of Mathematics & Computer/Information Sciences
University of Hyderabad, Hyderabad-500046
INDIA

CERTIFICATE

This is to certify that the project work entitled "**Enhanced On-line authentication using Zero-knowledge protocol based on Virtual password** ", being submitted for the partial fulfillment of the requirement for the award of the degree **Master of Technology in Computer Science** is a bonafide work carried out by **Venu Manga** under my supervision at the University of Hyderabad.

M.RukmaRekha
(Project Supervisor)
Department of Computer &Information Sciences,
University of Hyderabad, Hyderabad-500046,
INDIA.

HEAD
Department of Computer &
Information Sciences (CIS),
University of Hyderabad,
Hyderabad-500046, INDIA.

DEAN
School of Mathematics & Computer/
Information Sciences (MCIS),
University of Hyderabad,
Hyderabad-500046, INDIA.

To,

My Family and Friends

Acknowledgments

I would like to express my sincere gratitude to **Mrs.RukmaRekha**, my project supervisors, for valuable suggestions and keen personal interest throughout the progress of my course of research. They encouraged, supported, corrected and guided me during the project. The project has been a learning and growing experience for me. It is a good piece of work done in his guidance and a great opportunity to learn many things.

I am extremely grateful to our Head of the Department, **Prof. Arun Agarwal**, for providing excellent computing facilities and a nice atmosphere for doing my project. I convey my heartfelt thanks to **AI Lab staff** for their help in completing the project work successfully.

I would also like to thank the **Open Source Community** who provided the free Software and documentation to work with.

I would like to take this opportunity to thank my friends who have been morale boosters for me, encouraged me to take up this course and supported me throughout this course with their love and affection. My special thanks to my wonderful parents who have always supported me in all my decisions.

Venu Manga

ABSTRACT

Enhanced On-line authentication using Zero-knowledge protocol based on Virtual password

The Internet is allowing greater flexibility in working environment and location, like business Transactions, Online shopping, Online Banking, academic and government networks, which together carry lots of different information and services, these most common form of application is done through the use of password. The password can be transferred via a secure channel, even though the security of an application approach is still vulnerable to attacks like phishing, key-logger and shoulder-surfing. At the same time, It is a big challenge to find the most secured authentication system which is accepted by the users while using static password. The present solution to solve these attacks by using Zero-knowledge protocol based on a Virtual password.

Table of Contents

Chapter 1

Introduction

The Authentication is the process of verification of the identity of a person or entity. It is the process of determining whether someone or something is, in fact, who or what it is declared to be. Authentication is part of most online applications. Before a user can access its email account, its online banking account or its favorite online shopping account, it has to identify and authenticate itself to the application. The most common form of authentication is done through the use of passwords.

Internet usage and online applications are experiencing spectacular growth. Worldwide, there are over a billion Internet users at present. A big reason for the success of the Internet is the simplicity that you can access the applications from anywhere. This growth in popularity has not gone unnoticed by the criminal element – the simplicity of the HTTP protocol makes it easy to steal and spoof identity. The business liability associated with protecting online information has increased significantly and this is an issue that must be addressed.

1.1 Present day methods

Passwords enjoy ubiquitous use for online authentication. Although many more secure (typically also more complex and costly) authentication protocols have been proposed, the use of passwords for Internet user authentication remains predominant. Due to the usability and ease of deployment, most financial transactions over the Internet are authenticated through a password. Hence passwords are a prime target of attackers, for economically-motivated exploits including those targeting online bank accounts and identity theft.

Users with important accounts on the Internet face many kinds of attacks, e.g., a user ID and password can be stolen and misused. There are many reports on thefts on

ATMs as well. The secure protocol SSL/TLS for transmitting private data over the web is well known in academic research, but most current commercial websites still rely on the relatively weak protection mechanism of user authentications via a plaintext password and user ID. Meanwhile, even though a password can be transferred via a secure channel, this authentication approach is still vulnerable to attacks as follows.

1.1.1 Phishing: Phishers attempt to fraudulently acquire sensitive information, such as passwords and credit card details, by masquerading as a trustworthy person or business in an electronic communication. For example: A phisher can set up a fake website and then send some emails to potential victims to persuade them to access the fake website. This way, the phisher can easily get a clear-text of the victim's password.

1.1.2 Password stealing Trojan: This is a program that contains or installs malicious code. There are many such Trojan codes that have been found online today, so here we just briefly introduce two types of them. Key loggers capture keystrokes and store them somewhere in the machine, or send them back to the adversary. Once a key logger program is activated, it provides the adversary with any strings of texts that a person might enter online, consequently placing personal data and online account information at risk. Trojan Redirector was designed to redirect end-users network traffic to a location to where it was not intended. This includes crime ware that changes host files and other Domain Name Service (DNS) specific information, crime ware browser-helper objects that redirect users to fraudulent sites, and crime ware that may install a network level driver or filter to redirect users to fraudulent locations.

1.1.3 Shoulder-surfing: Shoulder-surfing is a well-known method of stealing other's passwords and other sensitive personal information by looking over victims' shoulders while they are sitting in front of terminals. This attack is most likely to occur in insecure and crowded public environments, such as an Internet Café, shopping mall, airport, etc. It is possible for an attacker to use a hidden camera to record all keyboard actions of a user for both a computer and an ATM machine. Video of the user's actions on a keyboard can be studied later to figure out a user's password and ID.

1.2 Use of various secure mechanisms

Use of virtual keyboards can prevent the key logger attacks by enabling the user to enter a password without using the keyboard but still the system is open to attacks like phishing, shoulder surfing. Use of secure access images can prevent phishing attacks to an extent, but they are not safe from shoulder surfing.

1.3 The Challenge

The security of an application is always a tradeoff between a high level of security and more usability. The more security is added to an authentication system (pass phrases instead of passwords, multiple authentication tokens), the lower will be the acceptance rate of the users and the usability will decrease. It is a big challenge to find the most secure authentication system which is accepted by the users.

Users always want new applications and features with easy to use interfaces. At the same time they are worried about the increasing dangers like phishing, identity theft, spyware, malware, keyloggers, javascript attacks, and generally untrusted consumer platforms all make the traditional means of password based authentication ever more complicated.

1.4 The Solution

The present solution is using zero knowledge protocol on a virtual password scheme. A virtual password is a password which cannot be applied directly to the server but instead generates a dynamic password which is submitted to the server for authentication. A virtual password Pd is composed of two parts, a fixed alphanumeric F and a function B. Function B uses a set of inputs n ,where B(n) = R. A function P is used to combine both the fixed and the random part of the password . P(F,R) = Pd is the virtual password and B is the virtual function. On the server side, the server can also calculate Pd in the same way to compare it with the submitted password.

A zero-knowledge proof of knowledge is a two-party protocol between a prover and a verifier, which allows the prover to convince the verifier that he knows some secret values that satisfy a given relation (proof of knowledge property), without the verifier being able to learn anything about them (zero-knowledge property). This can be done by the verifier asking various questions about the knowledge in a way which does not reveal the knowledge itself or the verifier may want the prover to perform some function for

some challenge value given by the verifier. Either way the verifier cannot learn anything about the knowledge itself. Here the knowledge which needs to be verified is the virtual password.

The user selects a function while registering whose input values change based on number of the visit. This function result will be our virtual password. Entering the virtual password directly could reveal function easily so we are using zero knowledge protocol on top of the virtual password where in the user will be provided with a challenge value for which the user has to perform a corresponding function on the previously computed virtual password. This result value will be finally sent to the server for authentication. We are trying to keep the function simple enough for usability and at the same time secure.

Chapter 2

Authentication Architecture

In this chapter we will discuss in depth about the Architecture of the authentication model and how this authentication model will work! How it will store the all information about the users! How it will identify the users, if that user is authorized users or unauthorized users!

Authentication is the process of determining whether someone or something is, in fact, who or what it is declared to be. In private and public computer networks and the Internet, authentication is commonly done through the use of logon passwords. Knowledge of the password is assumed to guarantee that the user is authentic. Each user registers initially or registered by someone else, using an assigned or self-declared password. On each subsequent use, the user must know and use the previously declared password. For example, a party wishing to admit a letter into evidence may ask the witness whether it is, indeed, the letter he received, does he recognize the handwriting, and similar questions. The weakness in this system for transactions that are significant (such as the exchange of money) is that passwords can often be stolen like phishing, key logger and shoulder-surfing attacks. For this reason, Internet business and many other transactions require a more stringent authentication process which is nothing but the password should not be static, it always change at the time of authentication. By means the password should be generated dynamically at the time of login and instead of entering the whole dynamic password, the server will ask some question to the user if the user know the details then only the user will give the correct response to the server then the server identifies the user is valid otherwise the user is invalid.

The present invention is related to communication networks and, in particular, to providing secure communications therein for providing access from a client computer over an insecure public network to one of a plurality of destination servers on a secure private network.

Computer networks are known generally as including a wide variety of computing devices, such as client computers and servers, interconnected by various connection

media. In particular, it is common place for an institution, such as a corporation, to provide such a network. Such network may include a multiplicity of servers executing a corresponding number of application programs ("applications"). The corporation's employees may use one or more of these applications to carry out the business of the corporation. Such a network may be characterized as a private, secure network, since it is accessible under normal, expected operating conditions only by suitably authorized individuals.

2.1 Authentication Architecture Description

In Authentication system when the user want to authenticate with the system, the architect need to provide intelligently controlling access to computer resources, enforcing policies, auditing usage, and providing the necessary information. These all services effectively performed by Authentication, authorization, and accounting (AAA) are a term of a framework. These combined processes are considered important for effective network management and security.

As the first process, authentication provides a way of identifying a user, typically by having the user enter a valid user name and valid password before access is granted. The process of authentication is based on each user having a unique set of criteria for gaining access. The AAA server compares a user's authentication credentials with other user credentials stored in a database. If the credentials match, the user is granted access to the network. If the credentials are at variance, authentication fails and network access is denied.

Following authentication, a user must gain authorization for doing certain tasks. After logging into a system, for instance, the user may try to issue commands. The authorization process determines whether the user has the authority to issue such commands. To put simply, authorization is the process of enforcing policies: determining what types or qualities of activities, resources, or services a user is permitted. Usually, authorization occurs within the context of authentication. Once you have authenticated a user, they may be authorized for different types of access or activity.

The final process in the AAA framework is accounting, which measures the resources a user consumes during access. This can include the amount of system time or the amount of data a user has sent and/or received during a session. Accounting is carried

out by logging of session statistics and usage information and is used for authorization control, billing, trend analysis, resource utilization, and capacity planning activities.

2.2 Authentication Model

Authentication model describes when the user want to authenticate with the system, the user need to interact with host application. The host application is an application that integrates with the development framework that allows you to develop your own application by using the application program interface and provides implementation of the service provider interface. The user interacts with host application with the help of user-interface; user-interface will provide the friendliness to the user in order to interact with the system. When the user communicates with the host application it invokes the application program interface. This application program interface is an interface defined and implemented by the framework that is used by host application to invoke operations provided by framework .Service provider interface is an interface defined by the framework that is implemented by host applications. An application will implement a number of service provider interfaces, in order to support the framework at the various support levels. Service provider interface will transfer the service to the application server. An application server is the kind of software engine that will deliver various applications to another device. It operates between the client and the database. It is the kind of computer found in an office or university network that allows everyone in the network to run software off the same machine.

An authentication mechanism using different types of components for establishing the secure connection between the valid user and application system, Authentication model components are

- User
- Host Application
- User Interface
- Application Program interface
- Service Provider Interface
- Application Server
- Database

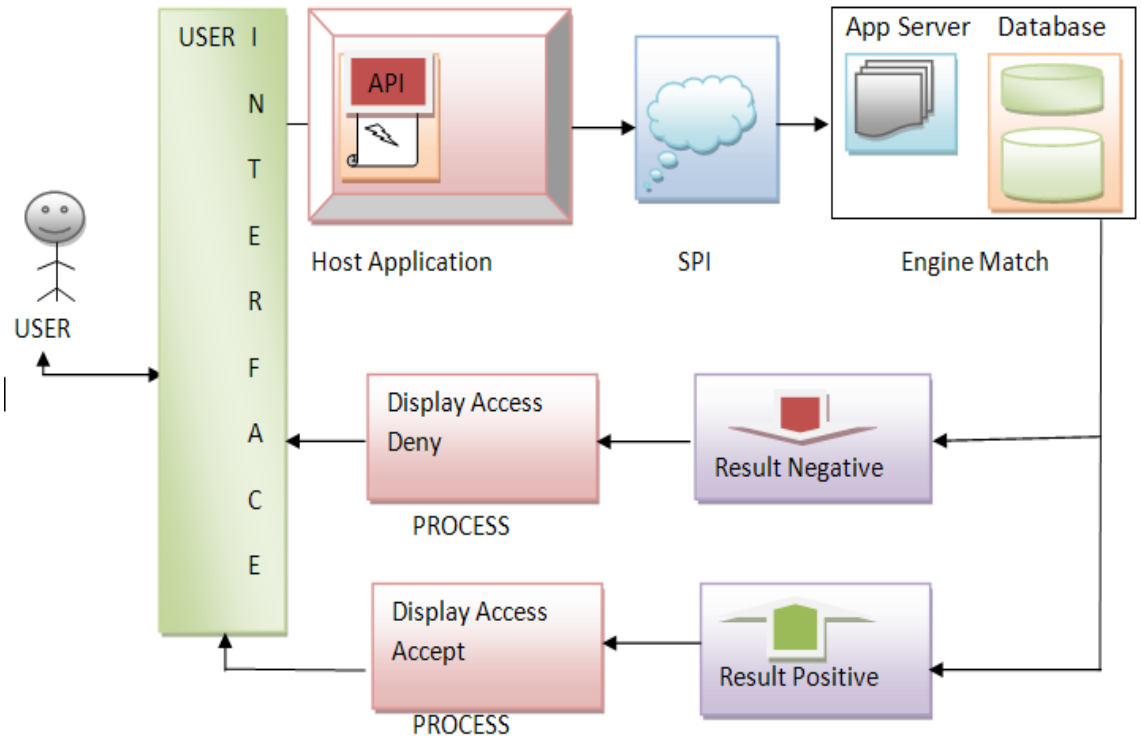

Figure2.1: Authentication Model

1. **User :** User is a person who makes use of a thing; someone who uses something entity that has authority to use an application, equipment, facility, process, or system, or one who consumes or employs a good or service to obtain a benefit or to solve a problem
2. **Host Application:** The host application is a type of device or system that allows individuals and organizations to make their own application accessible via the World Wide Web or without the Web.
3. **User Interface:** A user interface is the means through which a person controls a software application or hardware device. Application software is computer software designed to help the user perform a particular task. A good user interface provides a "user-friendly" experience, allowing the user to interact with the software or hardware in a natural and intuitive way. For example, nearly all

software programs have a graphical user interface, or GUI. This means the program includes graphical controls, which the user can select using a mouse or keyboard.

4. **Application Program interface:** An Application Program Interface(API) is an Interface which is used for accessing an application or a service from a program. An Application Program Interface makes it possible to use programs from within programs, therefore it is the foundation for modular systems with clearly defined interfaces between separate components. In a way, an Application Program Interface can be regarded as the local equivalent of a protocol, because it is used for the same purposes and defines the same things (the possible interactions between components, and the data that is exchanged while interacting). However, traditionally Application Program Interfaces are used for interfaces on one computer.

   An application program interface is the specific method prescribed by a computer operating system or by an application program by which a programmer writing an application program can make requests of the operating system or another application. An API can be contrasted with a graphical user interface or a command interface (both of which are direct user interfaces) as interfaces to an operating system or a program.

5. **Service Provider Interface:** Service provider interface is an interface defined by the framework that is implemented by host applications. An application will implement a number of service provider interfaces, in order to support the framework at various support levels. Services are units of functionality that are automatically available when an application program makes use of an implementation of the Application Program Interface.

   For example Services are the units of sound-handling functionality that are automatically available when an application program makes use of an implementation of the Java Sound API. They consist of objects that do the work of reading, writing, mixing, processing, and converting audio. An implementation of the Java Sound API generally supplies a basic set of services, but mechanisms

are also included in the API to support the development of new sound services by third-party developers (or by the vendor of the implementation itself). These new services can be "plugged into" an existing installed implementation to expand its functionality without requiring a new release. In the Java Sound API architecture, third-party services are integrated into the system in such a way that an application program's interface to them is the same as the interface to the "built-in" services.

Examples of potential third-party, sampled-audio services include:

- Sound file readers and writers
- Converters that translate between different audio data formats
- New audio mixers and input/output devices, whether implemented purely in software, or in hardware with a software interface

6. A**pplication Server:** Application servers manage the flow of data and the presentation of the data as it flows between the clients and the database server. They also can do much more to enhance efficiency, management, and the ability to expand client services. A simple example of an application server is a web server. It makes data available to the clients by processing it for the client. One purpose of application servers is to allow application program capabilities to be shared in an efficient and well structured manner.

6.1 Application Server model Description:

The application server may be the same computer as the web server or it may be separate from the web server. The application server also called an information server supports a distributed computing three tier model operating in the middle layer. It operates between the client and the database. The application server may also provide some subset of the following services, or provide program APIs to these services.

An application server is to allow application program capabilities to be shared in an efficient and well structured manner. Application servers should have the ability

to allow systems administrators rather than developers to deal with capacity management capabilities. In addition to program development and serving web pages to clients, application servers address the following issues which include in this model.

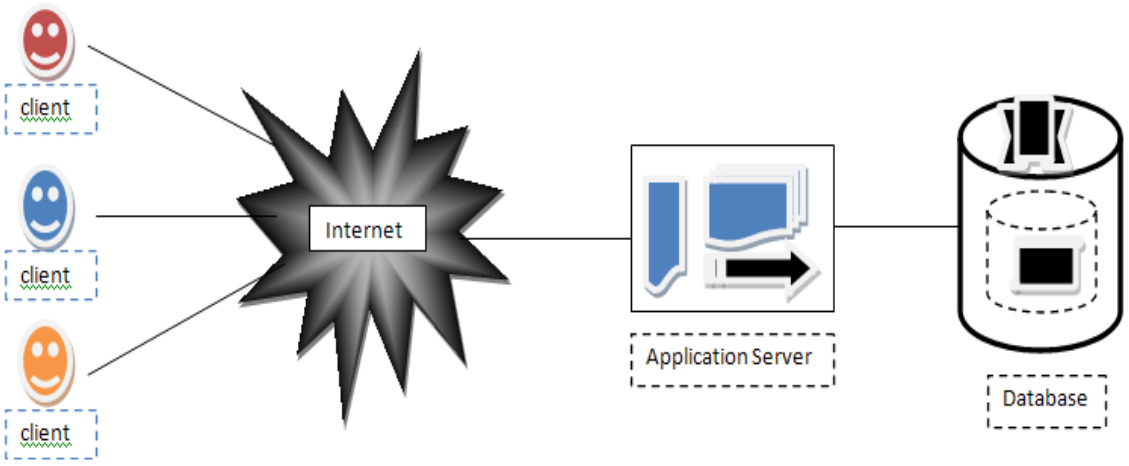

Figure2.2: Three Tier Model

First tier: Responsibility for presentation and user interaction resides with the first-tier components. These client components enable the user to interact with the second-tier processes in a secure and intuitive manner.

Clients do not access the third-tier services directly. For example, a client component provides a form on which a customer orders products. The client component submits this order to the second-tier processes, which check the product databases and perform tasks needed for billing and shipping.

Second tier (application logic layer): The second-tier processes are commonly referred to as the *application logic layer*. These processes manage the business logic of the application, and are permitted access to the third-tier services. The application logic layer is where most of the processing work occurs. Multiple client components can access the second-tier processes simultaneously, so this application logic layer must manage its own transactions.

In the previous example, if several customers attempt to place an order for the same item, of which only one remains, the application logic layer must determine who has the right to that item, update the database to reflect the purchase, and inform the other customers that the item is no longer available. Without an application logic layer, client components access the product database directly. The database is required to manage its own connections, typically locking out a record that is being accessed. A lock can occur when an item is placed into a shopping cart, preventing other customers from considering it for purchase. Separating the second and third tiers reduces the load on the third-tier services, supports more effective connection management, and can improve overall network performance.

Third tier: The third-tier services are protected from direct access by the client components residing within a secure network. Interaction must occur through the second-tier processes.

All three tiers must communicate with each other. Open, standard protocols and exposed APIs simplify this communication. These clients run on any operating system, by speaking with the application logic layer. Databases in the third tier can be of any design, if the application layer can query and manipulate them. The key to this architecture is the application logic layer.

7) **Database:** A database is a collection of data organized in a particular way. Databases can be of many types such as Flat File Databases, Relational Databases and Distributed Databases etc.

Whenever you create a database check whether it's a single table or a collection of tables, you need to look at the information you want to store and the ways you want to retrieve that information before you start working on the system. That's because a poorly structured database will hamstring you further down the track when you try to get your information back out in a usable form.

A database program gives you the tools to:

- design the structure of your database
- create data entry forms so you can get information into the database
- validate the data entered and check for inconsistencies
- sort and manipulate the data in the database
- query the database (that is, ask questions about the data)
- Produce flexible reports, both on screen and on paper, that make it easy to comprehend the information stored in the database.

Chapter 3

Related Work

In this chapter we will discuss in depth about the authentication mechanisms that are existing and the attacks that will occurs when the user want to authenticate with the system, and finally we discussed why username and password authentication play important role than all the existing authentication mechanisms and how we provide easy authentication mechanism to the user without loss of security and without the use of external hardware for authentication.

Networks, both internet and intranet, provides amazing opportunities, but not without some risk. Without the proper controls, your data is subject to several types of attacks. These problem areas are explained in the section. The different techniques are available for handling these attacks and we are still using Username and password authentication for the authentication and also we discuss why Username and password authentication technique play vital role than the existing security system in the authentication process.

Now a days most financial institutions offer their customers money transfers and other financial services (from account consultation to account aggregation, bills payment, international funds transfer, etc.) through the Internet channel. Online banking has quickly become an integral part of financial institutions' overall strategy. The number of online-banking users has tremendously increased and should keep on increasing in the years to come. By accessing banking services from any place and at any moment, end-users can benefit from increased convenience, simplicity and fastness. Besides, banks can reduce their transaction costs as e-banking is cheaper than traditional banking ways. They can strengthen their core business and broaden their customer scope by reaching valuable customers, selling new financial products and using e-services as an attractive differentiating tool.

However, with the fast growth of online fraud such as phishing and viruses attacks, Shoulder-Surfing, Internet banking comes with risks. Fraud and piracy constitute an undeniable threat that can depreciate the value added of such a service and increase consumers' reluctance to use e-banking services. An inescapable stake for banks

has emerged: creating and strengthening end-users' trust by guaranteeing security. As a consequence, financial institutions are now facing significant business challenges and have to define the most appropriate compromise between security, Convenience and Cost.

Customers must trust their financial institution and be certain that nobody will be able to access sensitive information or make transactions on their behalf. Passwords are no longer secure enough and costly to maintain and
manage. Making it more difficult to gain unauthorized access than static passwords, dynamic passwords - One-Time-Password-based schemes ensure the highest levels of security for online transactions. They offer the best compromise between security, convenience and business requirements, i.e. total cost of ownership, linked to large customer deployments.

3.1 Loss of Privacy

Phishing is the criminally fraudulent process of attempting to acquire sensitive information such as usernames, passwords and credit card details by masquerading as a trustworthy entity in an electronic communication. Communications purporting to be from popular social web sites, auction sites, online payment processors or IT Administrators are commonly used to lure the unsuspecting. Phishing is typically carried out by e-mail or instant messaging, and it often directs users to enter details at a fake website whose look and feel are almost identical to the legitimate one. Even when using server authentication, it may require tremendous skill to detect that the website is fake.

A perpetrator may observe confidential data as it traverses the Internet. This ability is probably the largest inhibitor of business-to-business communications today. Without encryption, every message sent may be read by an unauthorized party.

3.2 Loss of Data Integrity

Loss of Data integrity is known as "key-logger". "Key logging" is a shorthand term for the practice of keystroke logging. This is a form of computer surveillance that involves the recording of every key struck on a keyboard by software running locally (usually hidden) on the machine being monitored, as a hardware addition to the machine (such as a USB keystroke logger), or on the network that machine is connected to.

3.2.1 Software Keystroke Logging

Software keystroke loggers usually run as applications on the system that is being monitored and basically perform like any other piece of software on that operating system. The one quality that sets most keystroke loggers apart from other software is that programmers usually go to great lengths to hide evidence that the keystroke logger is actually running. This includes making the application invisible and usually only accessible by a predefined combination of keystrokes and obfuscating the process names to avoid identification.

There are several different types of software keystroke loggers. Hyper-visor based keystroke loggers run like virtual machines without employing the functions of the operating system. This makes them more difficult to detect. Kernel based keystroke loggers reside at the kernel level and are the most powerful keystroke loggers. Kernel based keystroke loggers subvert the operating system and gain direct access to the hardware through methods like spoofing the keyboard driver. Hook based keystroke loggers record keystrokes through functionality provided by the operating system. The passive method gains access to keystrokes through the operating system's APIs. And the Form Grabber method, perhaps the most common type of keystroke logger, records information submitted to web-forms. In some cases, as with many malware form grabbing keystroke loggers, information that is recorded can be transmitted to a remote computer for review.

3.2.2 Hardware Keystroke Logging

The key feature of hardware key loggers is that they do not depend on any software being installed. Hardware keystroke loggers come in two basic forms: firmware-based and as physical hardware. Firmware keystroke loggers work at the BIOS level and must be installed physically on the machine. This method works by adding a logging feature to the physical keyboard interface.

Hardware can also be installed directly in a computer keyboard, the computer's internal memory or a USB port in order to record keystrokes. The primary advantage to hardware keystroke loggers is that they do not depend on any software being installed on

the computer. This allows them to function with greater stealth and keeps them from interfering with other software on the computer. For example, a hook based software keystroke logger directly logs keystrokes through functions built into the operating system. However, this uses a notable amount of resources and a suspicious user could easily determine that some foreign software was running. Another advantage of high end hardware keystroke loggers is that they can interface with many computers over a network, providing wide reaching network surveillance.

Even for data that is not confidential, one must still take measures to ensure data integrity. For example, you may not care if anyone sees your routine business transaction, but you would certainly care if the transaction were modified. For example, if you were able to securely identify yourself to your bank using digital certificates, you would still want to ensure that the transaction itself is not modified in some way, such as changing the amount of the deposit.

3.3 Identity Spoofing

Moving beyond the protection of data itself, you must also be careful to protect your identity on the Internet. A crafty intruder may be able to impersonate you and have access to confidential information. Many security systems today rely on IP addresses to uniquely identify users. Unfortunately this system is quite easy to fool and has led to numerous break-ins.

3.4 Authentication Methods

The Authentication is the process of verifying the identity of a person or entity. It is the process of determining whether someone or something is, in fact, who or what it is declared to be. Authentication is part of most online applications. Before a user can access its email account, its online banking account or its favorite online shopping account, it has to identify and authenticate itself to the application.

Authentication technology provides the basis for access control in computer systems. If the identity of a user can be correctly verified, legitimate users can be granted access to system resources. Conversely, those attempting to gain access without proper authorization can be denied. Authentication is defined as the act of verifying the identity of a user. Once a user's identity is verified, access control techniques may be used to

mediate the user's access to data. A variety of methods are available for performing user authentication.

The traditional method for authenticating users has been to provide them with a secret password, which they must use when requesting access to a particular system. Password systems can be effective if managed properly (Federal Information Processing Standard [FIPS] 112), but they seldom are. Authentication which relies solely on passwords has often failed to provide adequate protection for computer systems for a number of reasons. If users are allowed to make up their own passwords, they tend to choose ones that are easy to remember and therefore easy to guess. If passwords are generated from a random combination of characters, users often write them down because they are difficult to remember.

Where password-only authentication is not adequate for an application, a number of alternative methods can be used alone or in combination to increase the security of the authentication process. The three generally accepted methods for verifying the identity of a user are based on something the user knows, such as a password; something the user possesses, such as an authentication token; and some physical characteristic of the user, such as a fingerprint or voice pattern. A variety of methods are available for performing authentication.

1) Password Authentication
2) Biometric Authentication
3) Security Token Authentication
4) Smart Card Authentication

3.4.1 Password Authentication

In most enterprises, the use of passwords is the primary means of authenticating a user. Unfortunately, it is also the weakest form of authentication. In today's digital world, the ways to bypass this form of security are trivial. While many enterprises focus on strengthening passwords, these efforts are by and large meaningless in the face of the tools that attackers can use. The tools provide criminals with easy ability to hack, trap, or crack most passwords easily.

The first attack tool against password authentication is a hardware keyboard logger. Legally available online for $40, these devices plug into the connection between the keyboard and the computer. They record every keystroke, with some models able to do time and date stamps against the data. A hardware keyboard logger looks like a small hardware piece of computer connections, takes only 10 seconds to install and is not detectable by any means of commercially available software.

The use of password authentication is further weakened by software attacks. This year alone, it is estimated that there will be several thousand different malware password logging attack programs will be created. Some of these are very sophisticated and can be ordered by the internet to attack certain types of firewalls. These password authentication logging software programs are embedded in email that are activated by clicking on the links in the email or by visiting a fake site that looks like the normal commercial site (phishing attack).

Some of the password authentication attacks are so sophisticated that there embed themselves on the core root operating systems kernel (rootkit attacks). Rootkit attacks are now acknowledged by Microsoft to be so insidious that the only way to remove them is to re-image every computer on the infected enterprise network!

3.4.2 Authentication - Biometrics

Biometric authentication is the process of verifying if a user or identity is who they claim to be using digitized biological pieces of the user. This can include finger scans, finger prints, iris scans, face scans, voice recognition and signature scans. Other biometrics in research for authentication includes vein scans and DNA.

Biometrics used for authentication is currently in fashion in the authentication industry. The UK and US governments are rapidly deploying them in their visas, passports and personal identification cards. Many other industries are adopting biometrics as authentication mechanisms for accessing bank machines, doorway access control and time card reporting and general computer desktop access. Authentication is the process of determining if a user or identity is who they claim to be. The authentication process is based on risk. Higher risk situations require more identity verification certainty. Biometrics can play a useful role in verifying the identity along with other factors.

Biometrics is very useful, in certain situations, as an authentication device. It is useful when someone is watching the user use a biometric authentication device. This way the enterprise can be relatively certain that there is no malfecance being done between the user, the biometric hardware reader and the enterprise security system. However, when biometrics are done remotely, with the enterprise not able to see and control the authentication hardware, the chances increase that the identity presenting their biometric may not be the person who is registered with the biometric. Therefore, the use of mutli-factor authentication mechanisms is used.

The use of biometrics as a deterrent against identity theft is being much touted at the moment. However, the use of biometrics alone will not likely deter criminals from finding ways around the use of biometrics. Remember that what is being presented are a set of computer bits that represent the biometric to the authentication server. Therefore, it is extremely likely that criminals will adjust their attack vectors and try to capture the biometric from the person, and then replay these on the enterprise.

3.4.3 Authentication - Security Tokens

Authentication is achieved by asking something you know, something you have or, providing something you are or combinations thereof. Something you have, like a physical token, is used often in real life e.g. a driver's license. In the digital world security tokens are now commonly used. They are often one time password security tokens and/or smart cards.

One Time Passwords

One time password security tokens, like secureID by RSA, are one way of significantly reducing the risk of using passwords. Unlike passwords which are changed every 60-90 days or longer, a secureID token works differently. On the small screen of the key fob the user carries with them are numbers that change every 60 seconds. The numbers displayed on the screen change randomly to the end user. They are generated by a mathematical algorithm that is only known to the enterprise security server.

The user logs on to the enterprise network. During the logon sequence the user is requested to enter in their id and then the number displayed on the screen. This information is sent via encryption to the enterprise security server. If the number on the screen matches the mathematical algorithm and the id, then the user is authenticated.

The devices are tamper proof/resistant. They are pre-programmed from the factory and ready for immediate use. By combining a secret that the user knows (their id) with the one-time password, the authentication is much stronger than that from a traditional password.

Authentication Weaknesses with Security Tokens

There are weaknesses with using only this approach. For instance, is someone is able to steal or fraudulently obtain the key fob and, they also know the user's id, then they will be able to successfully masquerade as the identity. Additionally, there are significant management costs with the key fobs or credit card size tokens. Recent announcements in February 2007 by Entrust selling one-time password tokens at $5 means that the price points are now much lower and more affordable. Users need to be issued them physically, they need to be replaced when lost (which is common) and recovered or terminated when an identity leaves the enterprise. Poor de-provisioning processes may result in security holes being created by the identity still having access to the network using their secureID token and id.

3.4.4 Access Control Cards - Contact less Smart Card

The contact less smart card has a microchip embedded in the card with internal memory. This enables the card to:

(a) Securely manage, store and offer data access to the card

(b) Perform complex functions and calculations (e.g. encryption)

(c) Interact with an RF device in an intelligent manner

Common applications of contact less smart cards include:

Mutual authentication:

The contactless smart card can verify that the card reader is authentic and then verify itself to the card reader before starting a secure transaction

Strong information security:

The ability of the microchip and memory enable the card to encrypt any identity information contained in the card as well as encrypting the RF connection between the contact less smart card and the card reader.

Tamper resistant security:

There are a number of hardware and software capabilities that is built into contact less smart cards to detect and react to tamper methods and help counter attacks on the card.

3.5 Authentication and Authorization Information Access Control

The contact less smart card can protect the information contained within the card by authenticating the information requestor and then allowing only the release of information the requestor is authorized for. The card owner may have additional methods such as a PIN number or a biometric to approve release of the information. This is an example of strong authentication.

As the enterprise risk rises for networks, applications and information access, so too must the layers of authentication strength. The financial system, payroll and payables are all higher risk. So too are users who hold super-user privileges like senior network administrators. For all of the medium and higher risk applications, your enterprise should be using a graded series of stronger authentication. For instance, low to medium risk might be addressed by the user providing their id, password and a digital certificate. Medium risk should be addressed by the user providing things like a secureID token along with their id and a password. Medium to high risk should be addressed by the user providing something like a smart card, a secure id token, a biometric and a second unique password.

The proliferation of networked computer systems and the corresponding increase in the potential for security violations makes it even more critical those who design and operate computer systems to understand and implement effective authentication schemes. There are many ways of authenticating a user. These range from the id and password (commonly referred to as "basic authentication"), digital certificates, security tokens, smart cards and biometrics. There are different reasons to use each type of authentication.

Why Password based authentication mechanism pre-dominant all the existing authentication mechanisms?

The Password-based authentication is the most widely used method for verifying the identity of persons requesting access to computer resources. However, authentication based only on passwords often does not provide adequate protection even though Password-based authentications enjoy ubiquitous use for online authentication. Although many more secure (typically also more complex and costly) authentication protocols have been proposed like authentication tokens, Smart cards, biometrics, and other alternative methods for verifying the identity of system users can substantially increase the security of an authentication system, the use of passwords for Internet user authentication remains predominant. Due to the usability and ease of deployment, most financial transactions over the Internet are authenticated through a password. Hence passwords are a prime target of attackers, for economically-motivated exploits including those targeting online bank accounts and identity theft. Due to this reasons we are proposing new algorithm zero-knowledge protocol based on virtual password, In this algorithm we are using virtual password mechanism and zero-knowledge protocol mechanism, by using this two mechanisms we are going to handle phishing, key-logger and shoulder-surfing attacks in effectively with out reducing the security in the authentication process to the user and at the same time we are trying to provide easy function to the user for the authentication. In this algorithm we two new concepts these are

- A virtual password is a password which cannot be applied directly but instead generates a dynamic password which is submitted to the server for authentication. On the server side, the server can also calculate Pd in the same way to compare it with the submitted password.
- A zero-knowledge proof of knowledge is a two-party protocol between a prover and a verifier, which allows the prover to convince the verifier that he knows some secret

values that satisfy a given relation (proof of knowledge property), without the verifier being able to learn anything about them (zero-knowledge property).

- The user selects a function while registering whose input values change based on number of the visit. This function result will be our virtual password. Entering the virtual password directly could reveal function easily so we are using zero knowledge protocol on top of the virtual password where in the user will be provided with a challenge value for which the user has to perform a corresponding function on the previously computed virtual password. This result value will be finally sent to the server for authentication. We are trying to keep the function simple enough for usability and at the same time secure.

We proposed a new algorithm how to prevent users' passwords from being stolen by adversaries. We introduced zero knowledge protocol based on virtual password concept involving a small amount of human computing to secure users' passwords in on-line environments and ATMs.

Chapter 4

The Zero-knowledge protocol based on Virtual password

In this chapter we will discuss in depth about Zero-knowledge protocol based on Virtual password Algorithm, and will present a brief discussion about Zero-knowledge protocol based on Virtual password.

A password is a sequence of characters obtained by a selection or generation process from a set of acceptable passwords. A good password system has a very large set of acceptable passwords in order to prevent an unauthorized person (or intruder) from determining a valid password in some way other than learning it from an authorized person. The set of acceptable passwords should be large enough to assure protection against searching and testing threats to the password system commensurate with the value of the data or resources that are being protected. The set of acceptable passwords must be such that it can be specified easily, that acceptable passwords can be generated or selected easily, that a valid password can be remembered, can be stored reasonably, and can be entered easily.

Users with important accounts on the Internet face many kinds of attacks, e.g., a user ID and password can be stolen and misused. There are many reports on thefts in online environment and ATMs as well. The secure protocol SSL/TLS for transmitting private data over the web is well known in academic research, but most current commercial websites still rely on the relatively weak protection mechanism of user authentications via a plaintext password and user ID. Meanwhile, even though a password can be transferred via a secure channel, this authentication approach is still vulnerable to attacks as phishing, keylogger, shoulder-surfing. Many schemes, protocols, and software have been designed to prevent users from some specified attacks. However, to the best of our knowledge, so far, there is not a scheme which can defend against all three types of attacks listed above at the same time. In this chapter, we present a password protection scheme that involves a small amount of human computing in an Internet-based environment, which will be resistant to a phishing scam, a Trojan horse, and shoulder-surfing attacks.

We propose a zero knowledge protocol based on virtual password concept that requires a small amount of human computing to secure users' passwords in on-line environments. users' passwords based on the fact that a server has more information than any adversary does. We analyze how the proposed scheme defends against phishing, Trojan horses, such as key loggers, and shoulder-surfing attacks.

4.1 Virtual password

To authenticate a user, a system (S) needs to verify a user (U) via the user's password (P) which the user provides. In this procedure, S authenticates U by using U and P, which is denoted as: S --> U: U, P. All of S, U, and P are fixed. It is very reasonable that a password should be constant for the purpose of easily remembering it. However, the price of easy to remember is that the password can be stolen by others and then used to access the victim's account. At the same time, we cannot put P in a randomly variant form, which will make it impossible for a user to remember the password. Virtual password is a password which cannot be applied directly but instead generates a dynamic password which is submitted to the server for authentication. A virtual password P is composed of two parts, a fixed alphanumeric F and a function B from the domain w to w, where the w is the letter space which can be used as passwords. We have P = (F,B) and B(F,R) = Pd, where R is a random number provided by the server (called the random salt and prompted in the login screen by the server) and Pd is a dynamic password used for authentication. Since we call P = (F,B) a virtual password, we call B a virtual function. The user input includes (ID, Pd), where ID is user ID. On the server side, the server can also calculate Pd in the same way to compare it with the submitted password. It is easy for the server to verify the user. The server can first find the user's record from the database based on the user' ID, and compute Pd, and compare it with the one provided by the user.

The virtual function plays a critical role in the virtual password. There are an infinite number of virtual functions, so that designing an appropriate function is very critical to the success of virtual password scheme. In order to defend against phishing, key-loggers, and shoulder surfing while the system is authenticating the user, this function should meet the following criteria:

The function should have some random input provided by the server, which then allows the users to type in different inputs each time they log in the system. This ensures the key logger cannot steal the password because the real password is not typed and the typed inputs change each time.

The function should be easy for the users. To make the system more secure, we could increase the complexity of the virtual function. However, this resulting function may be very difficult to remember or utilize. The objective is to design less complex but secure virtual functions.

The function should be unobservable, i.e., the observed password the user types in for the login session does not disclose hidden secrets; therefore, adversaries cannot use the stolen information to login to the system.

The function should be insolvable, i.e., the adversaries should not be able to solve the function with all the potential information they are able to obtain.

4.2 Zero-knowledge protocol

Zero-knowledge protocols allow identification, key exchange and other basic cryptographic operations to be implemented without leaking any secret information during the conversation and with smaller computational requirements than using comparable public key protocols.

Zero-knowledge protocols, as their name says, are cryptographic protocols which do not reveal the information or secret itself during the protocol, or to any eavesdropper. They have some very interesting properties, e.g. as the secret itself (e.g. your identity) is not transferred to the verifying party, they cannot try to masquerade as you to any third party.

Although Zero-knowledge protocols look a bit unusual, most usual cryptographic problems can be solved by using them, as well as with public key cryptography. For some applications, like key exchange (for later normal cheap and fast symmetric encryption on the communications link) or proving mutual identities, zero-knowledge protocols can in many occasions be a very good and suitable solution.

4.2.1 The parties in a Zero-knowledge protocol

The following people appear in zero-knowledge protocols:

Prover

Prover has some information that she wants to prove to Verifier, but she doesn't want to tell the secret itself to Verifier.

Verifier

Verifier asks Prover a series of questions, trying to find out if Prover really knows the secret or not. Verifier does not learn anything of the secret itself, even if he would cheat or not adhere to the protocol.

Eavesdropper

Eve is listening to the conversation between Prover and Verifier. A good zero-knowledge protocol also makes sure that any third-party will not learn a thing about the secret, and will not even be able to replay it for anyone else later to convince them.

Malice

Malice is listening to the protocol traffic and maliciously sending extra messages and modifying or destroying messages. The protocol must be tamper-resistant to this kind of activity.

4.2.2 Zero-knowledge terminology

The secret means some piece of information, be it a password, the private key of a public key cryptosystem, a solution to some mathematical problem or a set of credentials. With Zero-knowledge protocols, the prover can convince the verifier that she is in possession of the knowledge, the secret, without revealing the secret itself, unlike e.g. normal user name-password queries the building of confidence in each iteration of the protocol.

Often the prover will offer a problem to the verifier, which will ask for one of the two or more possible solutions. If the prover knows the real solution to the hard mathematical problem, she is able to provide any of the solutions asked for. If she doesn't know the real solution, she can not provide all of the possible solutions, and if the verifier asks for one of the other solutions, she is unable to provide it, and will be found out.

4.2.3 Modes of Operation

The zero-knowledge protocols can be used in three main modes. These are interactive mode, parallel mode and off line mode.

Interactive: Where Prover and Verifier interactively go through the protocol, building up the certainty piece by piece.

Parallel: Where Prover creates a number of problems and Verifier asks for a number of solutions at a time. This can be used to bring down the number of interactive messages with a slow-response-time connection.

Off line: Where Prover creates a number of problems, and then uses a cryptographically strong one-way hash function on the data and the set of problems to play the role of Victor, to select a random solution wanted for each problem. She then appends these solutions to the message. This mode can be used for digital signatures.

4.3 The Zero-knowledge protocol based on Virtual password Algorithm

1) Begin
2) Take the input from the user .

   // Userid=: UserName

3) The server display the function information.
4) The function part(Fp) will have 4 sub functions

   //These are f0,f1,f2,f3. (.'. Ip = f0f1f2f3)

5) **Function form** f(n) == Rn operation Sn.

   //Rn is the Random number
   //Sn is the Secrete number

   These Operations and Random numbers will be generated dynamically at the time of login.
   Secrete number should be stored in the database at the time of Registration.

   Example: Sn=6, Sn=8, Sn=3 etc.

6) **For Operation** :

   Assume Operation precedence at the time of Registration.

   0 for + operation

1 for - operation
2 for / operation
3 for * operation

- Run the Random number generation function 4 times in order to generate 4 random numbers (Range 0 to 3).
- When the Random number generation completed.
- It will give 4 Random numbers.

7) **For Random number** :

- Run The Random number generation program to generate 4 random numbers (Range 0 to 9).
- When the Random number generation completed.
- It will give 4 Random numbers.

8) **Function calculation** :

- Function f(0)==Rn1 Op Sn
- Function f(1)==Rn2 Op Sn
- Function f(2)==Rn3 Op Sn
- Function f(3)==Rn4 Op Sn

9) For Virtual Password

- Virtual Password will have string part and function part.
- String part was select at the time of registration.

// It is also stored in the database at the time of registration.

- Function part==f(0)f(1)f(2)f(3) .
- Virtual Password==string f(0)f(1)f(2)f(3).

Example : String part= uni321,

Function part=923853('.' f(0)=9,f(1)=23,f(2)=8,f(3)=53)

- Virtual Password==univ321923853

Virtual password length is = 13. ('. Positions 0 to 12).

10) Server side Function:

- Password Positions(Pp):
- Run The Random number generation functions in order to generate four random numbers (Range 0 to 100).
- When the Random number generation completed.
- It will give 4 Random numbers.

  Ppi= Rni (modular Division) Vp length.

- Server do the calculation in order to generate the 4 positions.

  Example: Positions are 12, 8 , 4, 0.

11. **Authentication Checking**:

- Server will ask the character or integer at that positions to the User.

  If The User know the character or integer of that positions.

  Then Only

  The user who give the correct response
  And that user is the authorized user.

  Else

  The user who give the in valid response
  And that user is the un authorized user.

  End

4.3.1 Algorithm Description

An authentication model is store all the necessary information regarding the user in the database in order to find, who are authenticating with the system, that particular person/entity is valid or invalid. In our algorithm the user store the string part, secrete number and operation precedence in the database at the time of registration phase. Each user registers initially or registered by someone else, using assigned or self-declared

details. All information regarding the user will be stored more securely in the database. When the users want to authenticate with the system, the server will provide user login view for authentication. This user login web browser is act as a friendly environment to the user, by using this user interface, the user need to perform some calculations in the mean time the server will ask some number of question in sequence, if the user know the answer then only the user will give the correct response to the server for different type questions. In this way the authentication process will work, if the user will satisfies the server then only that particular person is valid user otherwise the user is invalid user.

The below user login view provided by the server when the users want to authenticate with the system. The user login interface will work, in order to provide sufficient information for user login. The user will enter the necessary details for login, if the user knows the correct details then only the user will enter the correct details otherwise the user will never enter correct details.

Figure 4.1: User Login View: First phase

In our algorithm when the user s want to authenticate with the system the user need to perform two steps of process where as in the first step, the user will enter is user

name in the login browser and then after the user will send request to the server. The server will verify the user request in the database, if the user name is available and valid in the database then only further details displayed to user in order to access the complete authentication. If the user name is not available in the database the user is invalid user and the permissions denied, to access the particular application for required second step process is denied itself in the first step of authentication process. If the user is valid in the first step of authentication process then only he/she get the details of second process, where as in the second process step the server displays the four random numbers and four operation precedence random numbers, here the random numbers and operation precedence numbers are randomly generated by the server at the time of login phase. These random numbers and operation precedence random details will be generated by the server, at each login phase in the user authentication system. Operator precedence will be stored in the database at the time of registration, based on user requirement from the existing simple mathematical functions. The existing simple mathematical functions such as addition, subtraction, multiplication, division, square of random number, square of secrete number, two to the power of random numbers, natural number (range is single digit like 0,1,2,……9) power of secrete number, cube function on the secrete number etc. The user will choose four functions based on his/her convince do some calculations for in order to generate virtual password. For these virtual password, the user need string part, secrete number including with random numbers and operator precedence of the functions, here string part, secrete number and operator precedence also securely stored in the database at the time of registration.

A virtual password is a password which cannot be applied directly to the server but instead generates a dynamic password with the help of above information. In our algorithm virtual password is sting part followed by function part. A virtual password consists of two parts one is string part and the second is the function part.

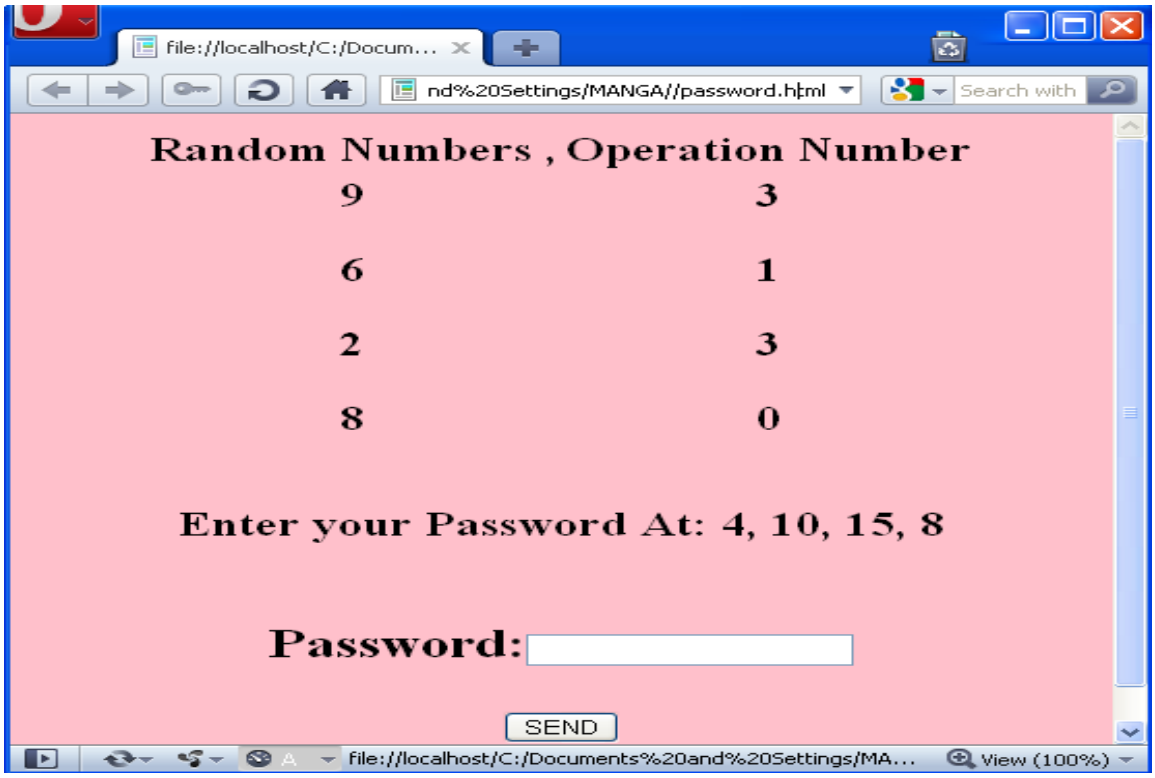

Figure 4.2: User Login View: Second phase

The string part is like a static password, it is a sequence of alphabetic, numeric numbers, special symbol, or combination of all these things. The string part is having at least six characters in the string for example : uni123, ven549, 497216, abcxyz,9030497216,9440405323 …etc. these all are the different type string parts and where as the function part is having four sub function based on secrete number, four random numbers and four operation precedence,

Function form f(n) == Rn operation Sn.

//Rn is the Random number
//Sn is the Secrete number

These Operations and Random numbers will be generated dynamically at the time of login.
Secrete number should be stored in the database at the time of registration. Example: Sn=6, Sn=8, Sn=3 etc.

Function calculation :

- Function f(0)==Rn1 Op Sn
- Function f(1)==Rn2 Op Sn
- Function f(2)==Rn3 Op Sn

- Function f(3)==Rn4 Op Sn

  Function part==f(0)f(1)f(2)f(3) .

Function part is nothing but the sequence of four functions. These function part f(n) is computed from the secrete number, four random numbers and four operator precedence random numbers provided by the server at the time of authentication.

Virtual Password:

A virtual password is a password which cannot be applied directly to the server but instead generates a dynamic password by using simple mathematical functions with the help of random numbers.Virtual Password will have string part followed by function part.

Virtual Password==string part & function part.

For example, One user stored the values in the database the values are string part is 497216, secrete number as 8 and operation precedence. By using secrete number, random numbers and operator precedence the user calculated function part.

Function part=923853

(‘.’ f(0)=9,f(1)=23,f(2)=8,f(3)=53)

Now the user is having string part and function part, in order to integrate this string part and function part in to single one is known as virtual password. Virtual password as string part followed by function part.

Virtual Password==string part & function part

Virtual Password==497216923853

In the above virtual password is having 12 characters, the virtual password length is always starting from 0. The number of characters present in the virtual password is 0 to 11. The virtual password length is always change, it is not fixed, and it is difficult to hackers in order to find the function position in the virtual password, and some what difficult to where the particular function is starting.

Virtual Password==497216923853

Virtual password length is = 12.

(.'. Positions 0 to 12).

A user has to remember both the fixed part and the function part, and as a result will require a little bit more effort to remember. However, the virtual password will be resistant to a dictionary attack, which is mostly caused by the fact that users like to create a password which is either related to their own name, date of birth, other simple words, etc.

The server then delivers this function information to the user via some channels, such as, displaying it on the screen or email. The user needs to remember this function together with the password they have chosen or save them in disks or emails. We also note that a small amount of human-computing is involved in the authentication process. In a traditional password scheme, users can change their passwords, and this fact is also true in our virtual password scheme. Different from the traditional scheme, users can change the string part of the virtual password, secrete number, or the operation precedence, or even all.

Server side Function:

In our algorithm when the users want to authenticate with the system the user will send request to the server. The server will verify the user request in the database, if the user name is available and valid in the database then only further details displayed to user in order to access the complete authentication. If the username is not available in the database the user is invalid user and the permissions denied, to access the particular application for required second step process is denied itself in the first step of authentication process. If the user is valid in the first step of authentication process then only he/she get the details of second process, where as in the second process step the server displays the four random numbers and four operation precedence random numbers, here the random numbers and operation precedence numbers are randomly generated by the server at the time of login phase. These random numbers and operation precedence

random details will be generated by the server, at each login phase in the user authentication system.

The client side how the users calculate the virtual based on the user secrete number, random numbers and operation precedence. In the same way, the server will also calculate the virtual password, for that particular user only based on that user stored values in the database. Now the server also having the same virtual password at the server side, instead of asking the whole virtual password for authentication, it will generate some random position within the virtual password range, for this one the server will generate four random numbers with in the range 0 to 1000, each random number would be modulo division with the virtual password length. When the server will modulo division with the virtual password length, then all the four position will be with in the range of virtual password.

Password Positions(Pp):

- Run The Random number generation functions in order to generate four random numbers (Range 0 to 100).
- When the Random number generation completed. It will give four Random numbers.

Ppi= Rni (modular division) Vp length.

Pp1=Rn1(modulo division) Vp length.

Pp2=Rn2(modulo division) Vp length.

Pp3=Rn3(modulo division) Vp length.

Pp4=Rn4(modulo division) Vp length.

In this way the server calculate the function in order to generate four random positions, Now the server is having four random positions for example: Positions are 12, 8 , 4, 0. The server will ask that positions of the characters at that particular positions.

Authentication Checking:

In authentication checking the server will take the response from the user in order to find the valid users, if the user knows the correct answer at that particular position then only the user will give the correct response to the server otherwise the user will not give the correct answer to the server. The user who give the correct response to the server is valid user otherwise the user is invalid user.

Chapter 5

Results and Discussion

In this chapter we will discuss in depth about how our Zero-knowledge protocol based on Virtual password Algorithm will work and how it will give the results, and will present a brief discussion about Zero-knowledge protocol based on Virtual password performance.

5.1 Experiments and Results

In order to implement the zero knowledge protocol based on virtual password scheme to safeguard users when they are surfing online, we implemented the scheme, and demonstrate that a little human computing can defeat phishing, key logger, and shoulder-surfing attacks. In this section, we will evaluate our practical implementation of the zero knowledge protocol based on virtual password scheme. This does not decrease the scheme's security due to the limited value of secrete number , since in the value of the dynamic password will rely heavily on the random number and order of preference in the operation selection.

Currently, most of the websites allow a user to have only one fixed unique password. In our scheme, however, the password is dynamic and a user needs to make some computations for each login, which is significantly different from the traditional way that the user just inputs a password. The traditional way may seem more comfortable to the user, but the price of such comfortableness is that the password could be stolen by adversaries. If considering the fact that users tend to pick passwords that are usually used in cross systems for easy recall, or those related to the users' privacy, such as DOB, nick name, and so on, the traditional password is more vulnerable.

Although it is tedious for users to make some calculations each time to login to the system, the well-trained user can finish the entire login process in a short time. We are trying to secure internet and most of them would accept the cost of spending a little more extra time to sign onto the system for an improvement in password security. We argue that the extra time will be acceptable to most of the people because the extra time will be very small, and there is no extra time at all if a user's mobile device can communicate with the server.

5.1.1 Usability test

We examined the users whether the approach is easily used by users or not. In order to test the usability of our scheme, we conducted a usability test. In our testing, each volunteer was asked to try to login to our test website for authentication.

1	User 01	User 02	User 03	User 04	User 05	User 06	User 07	User 08
2	41	50	28	42	34	18	31	44
3	44	54	33	27	26	29	26	39
4	30	28	21	38	41	21	38	28
5	13	33	18	30	16	34	41	34
6	21	48	35	26	33	16	19	38
7	40	39	26	21	38	40	25	31
8	27	46	21	39	31	26	21	21
9	40	58	38	44	22	32	40	42
10	35	19	12	12	43	17	18	29
11	29	26	24	29	19	21	29	20

Figure 5.1: User Log in time in seconds

In the above table we recorded login time for the different users, in seconds. For each user we examined ten times for authentication and we recorded in each authentication how much of time the user is taking. We examined around 70 users for authentication how much of time they are taking and whether this approach is easily used by users or not.

For this authentication, the users needed to calculate the password by themselves, with the help of their string part, secret number and operation precedence. We recorded the time how much time needed to complete the process. They completed each round 10 times and recorded the time it took them to complete their login. We can see that the user time to login to the system can vary depending on their ability to perform simple calculations. Some users will take around 25s to login to the system. Some other users will take around 50s to login the system.

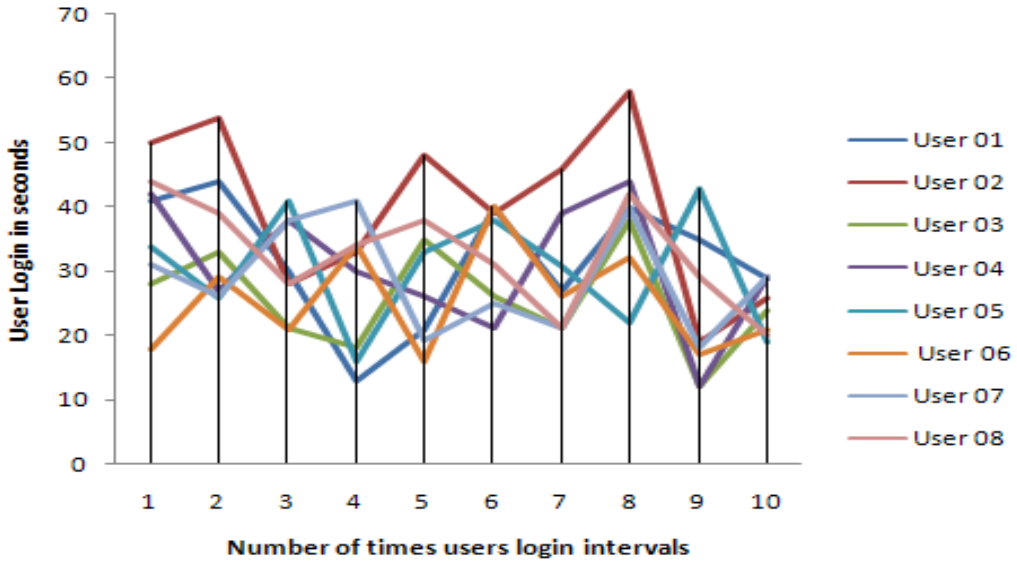

Figure 5.2 Log in time for Different Users

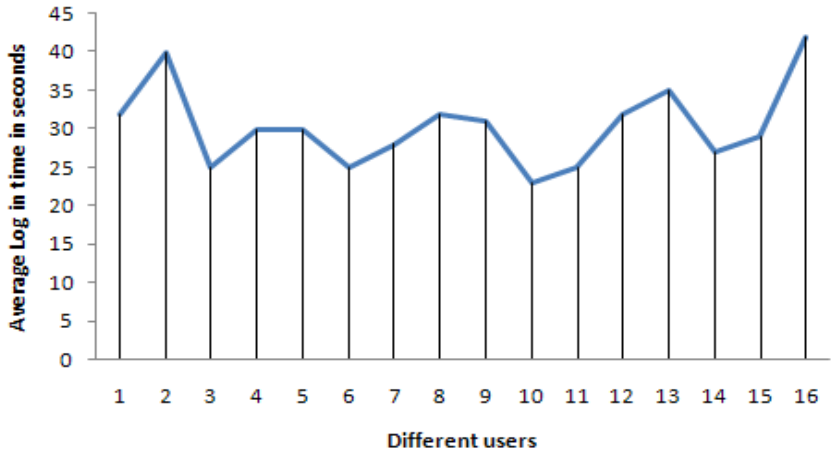

Figure 5.3: Average Log in Time

The total users will take an average of 35s to login to the system. We examined the users that it is to take a little bit longer to login to the system, if such a login will be guaranteed secure. The login success rate is around 90%. We argue that it is worth it to take a little bit longer to login to the system, if such a login will be guaranteed secure. This is especially important when a user logs into some important system via the internet, such as an online banking account or credit card account, online shopping and government networks.

Chapter 6

Conclusion and Future Work

Conclusion:

We proposed a new algorithm to prevent user's password from being stolen by adversaries. We introduced a new mechanism of authentication using zero knowledge protocol based on a virtual password involving a small amount of human calculation to secure user's password in online environments and ATM's. We analysed how the proposed scheme defends against phising, keylogger and shoulder surfing attacks.

Future work:

The future work of this authentication scheme involves in increasing the usability by decreasing the amount of human calculation involved and also increasing the security at the same time.

References

[1]. Ming Lie,Yang Xiao,Susan V. Vrbsky,Chung-Chih Li:Virtual password using random linear functions for on-line services, ATM machines, and pervasive computing. Computer Communications 31(18): 4367-4375 (2008)

[2]. C.Herley, D.Florencio, How to login from an Internet Cafe without worrying about Key-loggers, : in proceedings of Symposium on Usable Privacy and Security (SOUPS)'2006.

[3]. Modern access control based on eye movement analysis and keystroke dynamic'2006. Adrian Kapczynskil, Pawel Kasprowski, Piotr Kuzniacki.

[4]. Zero Knowledge Protocols and Small Systems -Hannu A. Aronsson, Department of Computer Science Helsinki University of Technology, haa@cs.hut.fi

[5] B. Ross, C. Jackson, N. Miyake, D. Boneh, J. Mitchell, Stronger password authentication using browser extensions, in: Proceedings of 14th USENIX Security Symposium.

[6] E. Gaber, P. Gobbons, Y. Mattias, A. Mayer, How to make personalized web browsing simple, secure, and anonymous, in: Proceedings of Financial Crypto'97, LNCS, vol. 1318, Springer-Verlag, 1997.

[7] E. Gabber, P. Gibbons, D. Kristol, Y. Matias, A. Mayer, On secure and pseudonymous user-relationships with multiple servers, ACM Transactions on Information and System Security 2 (4) (1999) 390–415.

[8] E. Damiani et al., Spam attacks: P2P to the rescue, in: Proceedings of Thirteenth International World Wide Web Conference, 2004, pp. 358–359.

[9] V.A. Brennen, Cryptography Dictionary, vol. 2005, 1.0.0 ed., 2004.

[10] M. Abadi, L. Bharat, A. Marais, System and method for generating unique passwords, US Patent 6 (141) (1997) 760.

[11] M. Kuhn, Probability theory for pickpockets – ec-PIN guessing, Available from: <http://www.cl.cam.ac.uk/?mgk25/>, 1997.

[12] A. Herzberg, A. Gbara, Trustbar: protecting (even naive) web users from spoofing and phishing attacks, Cryptology ePrint Archive, Report 2004/155, 2004. Available from: <http://eprint.iacr.org/2004/155/>.

[13] B. Sun, Y. Xiao, C.-C. Li, H.-H. Chen, T.A. Yang, Security co-existence of wireless sensor networks and RFID for pervasive computing, Computer Communications, Special Issue on Secure Multi-Mode Systems and their Applications for Pervasive Computing.

[14] T.A. Meyer, B. Whateley, SpamBayes: effective open-source, Bayesian based, email classification system, in: Proceedings of the CEAS, 2004.

[15] B. Sun, C.-C. Li, K. Wu, Y. Xiao, A lightweight secure protocol for wireless sensor networks, Computer Communications 29 (13–14) (2006) 2556–2568.